2.31 PROPORTIONAL NAV GUIDANCE

This document describes the command guided missiles in ESAMS.

SEE CLASSIFIED ADDENDUM

This document thoroughly describes command guidance

SEE CLASSIFIED ADDENDUM

Fundamentals of Command Guidance

The geometry between a target aircraft and a command guided missile is illustrated in Figure 2.31-1.

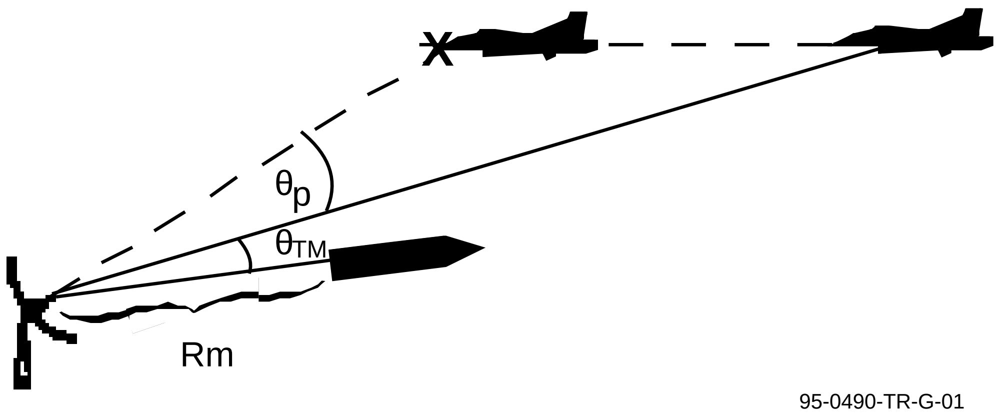


FIGURE 2.31-1. Command Guidance Geometry.

The missile can be in one of two modes: three point (beam rider) or lead angle. In the three point mode, the missile is commanded to ride the "beam" up to intercept of the target. Thus, the ground station sends commands to the missile that are designed to align the target aircraft, missile, and ground station. The commands are provided in two orthogonal directions, such as azimuth and elevation, so that intercepts can be made in three dimensional space.

Lead angle guidance may be employed if range to the target is known. If this parameter is available, the missile can be commanded to lead the target so that, based on "time-to-go," the target and missile are on a collision trajectory. This mode is attractive against non-maneuvering targets, since it minimizes the "Gs" which a missile must use in order to intercept the target. A weighting of .5 on the lead angle mode has been found to be optimal, since it reduces "Gs" against non-maneuvering targets and serves as a hedge against the target initiating maneuvers.

A minimal manner of modeling command guidance is illustrated in Figure 2.31-2, where s is the Laplace operator d/dt:

$$\text{Normal Accel}\Big|_{lim} = \frac{\overbrace{S.F.}^{\text{Stiffness Factor}} \cdot \left[\overbrace{R_m\dot\theta_{TM}}^{3PT} + .5\,R_m \cdot \overbrace{\left[\frac{\dot\theta_T\, R_{TM}}{\dot R_{TM}}\right]}^{\text{Lead Angle } \theta_p} \overbrace{+\,K\cdot}^{\text{Gearing Ratio}} \overbrace{\frac{(cmd_{pres} - cmd_{past})}{dt}}^{\text{Damping}}\right]}{\underbrace{1+\tau S}_{\text{System Lag}}}$$

95-0490-TR-G-02

FIGURE 2.31-2. Representative Command Guidance System.

The three point and lead angle commands move the missile through the desired lateral offset so that the missile achieves the desired alignment with respect to the target. The three point command is the simple product of missile range times angular offset, while the lead angle term requires missile range (R_M) times target angular rate $(\dot\theta_T)$ times "time-to-go" $(R_{TM}/\dot R_{TM})$, where R_{TM} is the range between target and missile and $\dot\theta_T$ times $R_{TM}/\dot R_{TM}$ is the predicted target position (θ_p) at intercept. Since the commands are in terms of distance (meters), a stiffness factor is applied so that the command is transformed into acceleration. The acceleration command is then processed by the system, with the achieved acceleration being limited by the maximum system capability.

Two more terms appear in figure 2.31-2. The first is gearing ratio times damping, and the second is the system lag. The first term is known as "lead compensation," and it is used for stability purposes. If the missile is approaching the desired position, then the present command (cmd pres) will be smaller than the past command (cmd past), and the damping term will oppose the primary command and thus impose a braking action. If the missile is receding from the desired point, then (cmd pres) will be greater than (cmd past), and the damping term will be additive to the primary command. Thus, in all cases, this term aids in meeting objectives, with the gearing ratio having been selected to provide the correct lateral response profile of the missile with respect to the target aircraft.

The time lag denoted in Figure 2.31-2 is a simple way to represent the lags inherent in generating the system response. It is known as a first order time lag. As will become evident, both first and second order responses are heavily used to represent dynamic systems such as SAMs.

Figure 2.31-3 contrasts first and second order responses. The second order response is very popular, since physical system motion can often be quantified in terms of damping ratio and natural frequency.

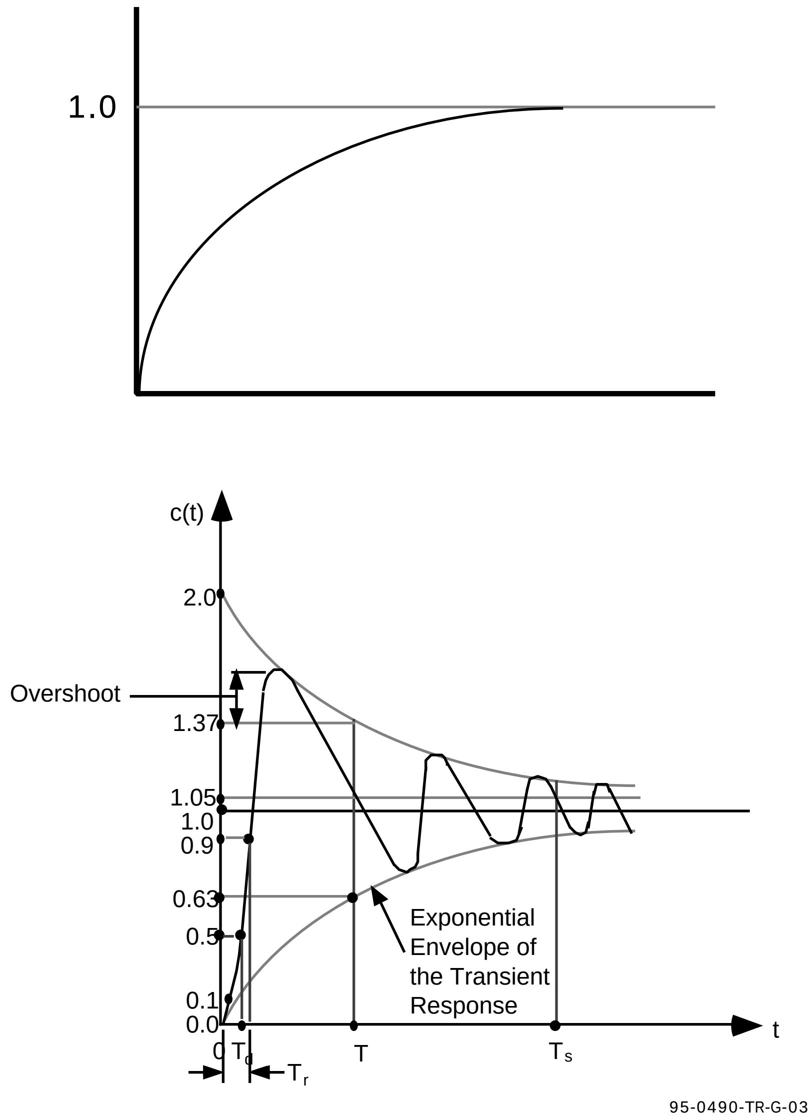


FIGURE 2.31-3. First and Second Order Responses.

Figure 2.31-4 represents a more detailed simulation. The incorporation of the comprehensive lead nets, lag nets, and autopilot allows the missile transient and steady state performance to be addressed in more detail. The function of the autopilot, of course, is to move the control surfaces in response to the guidance commands to steer the missile as desired. The achieved acceleration developed by the autopilot is fed into the element in figure 2.31-4 labeled dynamics, and the resultant lateral displacement of the missile is fed back and compared with the desired displacement. The process continues until the engagement terminates.

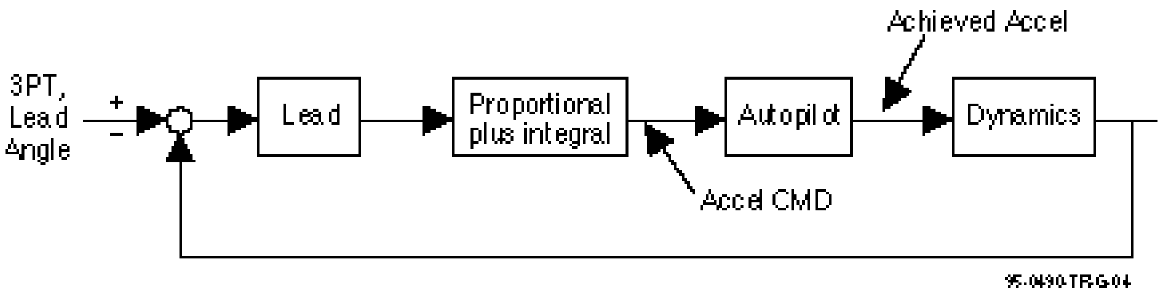


FIGURE 2.31-4. High Fidelity Command Guidance Model.

SA-8 Guidance Employment

In the last section, an overview of command guidance—

SEE CLASSIFIED ADDENDUM

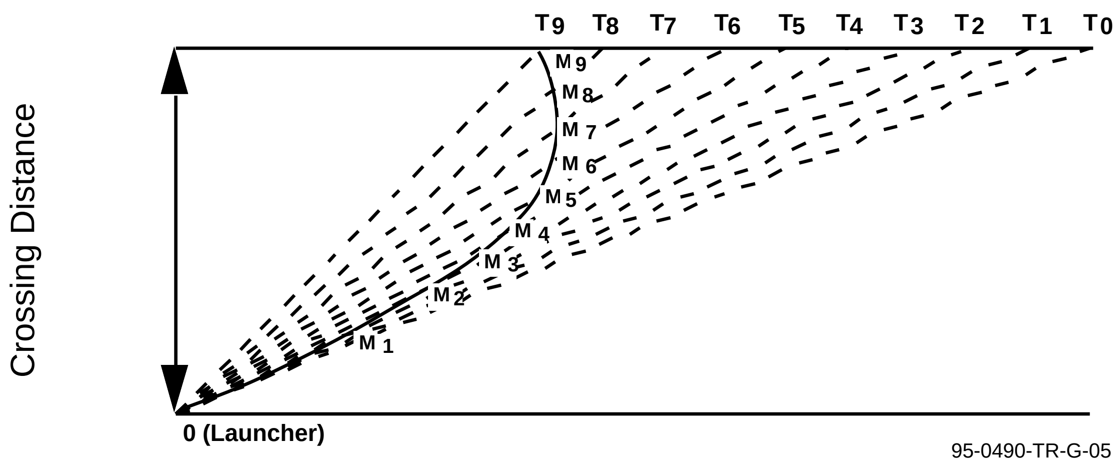


FIGURE 2.31-5. A Typical Missile Trajectory for an Approaching Target

In figure 2.31-5,

SEE CLASSIFIED ADDENDUM

As reviewed in Garnell and East (reference 2, pages 134-138), there are several issues to be addressed in developing dynamic compensation. A key consideration is the Coriolis component of acceleration that is experienced when a body is moving along a rotating line with a velocity of U. The Coriolis acceleration is $2U\dot{\theta}$, where $\dot{\theta}$ is the rotation rate of the target trajectory and therefore the desired rotation rate of the missile (figure 2.31-6). Thus, the guidance command must account for moving the missile onto the line-of-sight and also for the Coriolis acceleration necessary to fly on a curved trajectory.

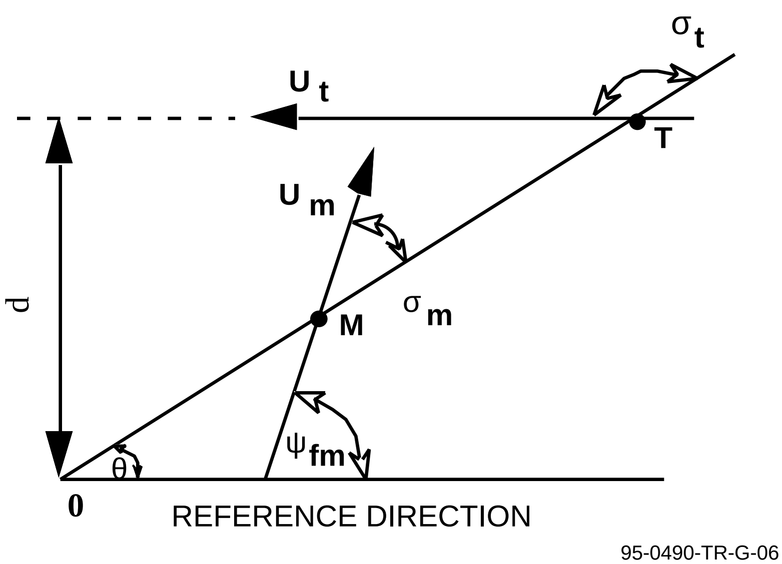


FIGURE 2.31-6. Target Aircraft/Tracking Radar.

As explained in reference 1, there are other undesired effects.

SEE CLASSIFIED ADDENDUM

There are some other signals that are also developed in

SEE CLASSIFIED ADDENDUM

2.31.1 Functional Element Design Requirements

This section contains the design requirements to fully implement **SEE CLASSIFIED ADDENDUM**:

1. **SEE CLASSIFIED ADDENDUM**.
2. **SEE CLASSIFIED ADDENDUM**.
3. **SEE CLASSIFIED ADDENDUM**.
4. **SEE CLASSIFIED ADDENDUM**.

These requirements must be implemented in sufficient fidelity to reflect the true missile intercept capability over

SEE CLASSIFIED ADDENDUM.

2.31.2 Functional Element Design Approach

This section describes the design approach (equations, algorithms, and methodology) implementing the design requirements of the previous section.

Figure 2.31.2-1 and 2.31.2-2

SEE CLASSIFIED ADDENDUM

Design Element 31-1: Injection Mode

SEE CLASSIFIED ADDENDUM

SEE CLASSIFIED ADDENDUM

Figure 2.31.2.1-1 indicates that the **SEE CLASSIFIED ADDENDUM**.

Early in the engagement, the missile will be in a wide or medium beam phase (see figure 2.31.2.1-2). In either of these phases, the missile is tracked by the missile tracker (MTR) while the target is tracked by the target tracker (TTR). Since the angle between MTR boresight and TTR boresight is known, the difference between the missile and target can be calculated as $\delta M_{\varepsilon} + \Delta\varepsilon_{M} - \delta T_{\varepsilon} =$ angular difference between missile and target in elevation.

SEE CLASSIFIED ADDENDUM

SEE CLASSIFIED ADDENDUM

SEE CLASSIFIED ADDENDUM

SEE CLASSIFIED

FIGURE 2.31.2.1-1. Guidance Azimuth Block Diagram

FIGURE 2.31-7. Wide/Medium Beam Angular Error.

FIGURE 2.31-8. Narrow Beam Angular Error.

SEE CLASSIFIED

FIGURE 2.31.2.1-4. Injection Mode.

$$\boxed{\frac{.165\,s + 1}{.0625 s^2 + .325\,s + 1}}$$

$$\Downarrow$$

$$\frac{y}{u} = \frac{.165\,s + 1}{.0625 s^2 + .325\,s + 1} = \frac{2.64\,s + 16}{s^2 + 5.2\,s + 16}$$

$$a_0 y^{(2)} + a_1 y^{(1)} + a_2 y = b_0 u^{(2)} + b_1 u^{(1)} + b_2 u$$
$$y = y_1 + K_0\,u$$

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{a_2}{a_0} & -\frac{a_1}{a_0} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} u$$

$$y = y_1 + K_0\,u$$

$$K_i = b_i - \sum_{m=0}^{i-1} a_{i-m}\,k_m$$

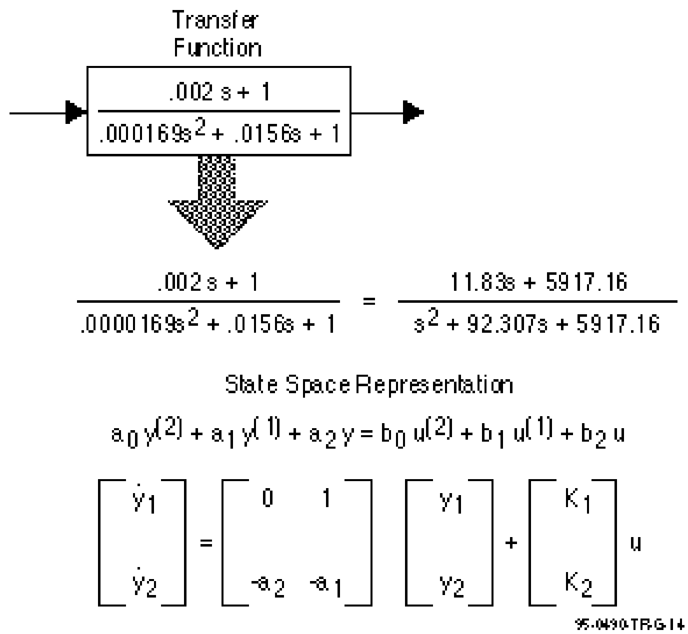
$$k_0 = b_0 = 0$$
$$k_1 = b_1 - a_1 k_0 = 2.64$$
$$k_2 = b_2 - a_2 k_0 - a_1 k_1 = 16$$

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -16 & -5.2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} 2.64 \\ 2.14 \end{bmatrix} u$$

95-04-90-TRG-13

The filtered target angle $\delta T_{\varepsilon 1}$ is with respect to the target tracker boresight. In either the wide or medium beam mode, this magnitude is differenced with the sum of the missile angle off missile tracker boresight (δ_{M_ε}) and the delta between missile tracker and target tracker boresights $(\Delta_{\varepsilon M})$.

SEE CLASSIFIED ADDENDUM:



$$K_i = b_i - \sum_{m=0}^{i-1} a_{i-m} K_m$$

$$K_0 = b_0 = 0$$

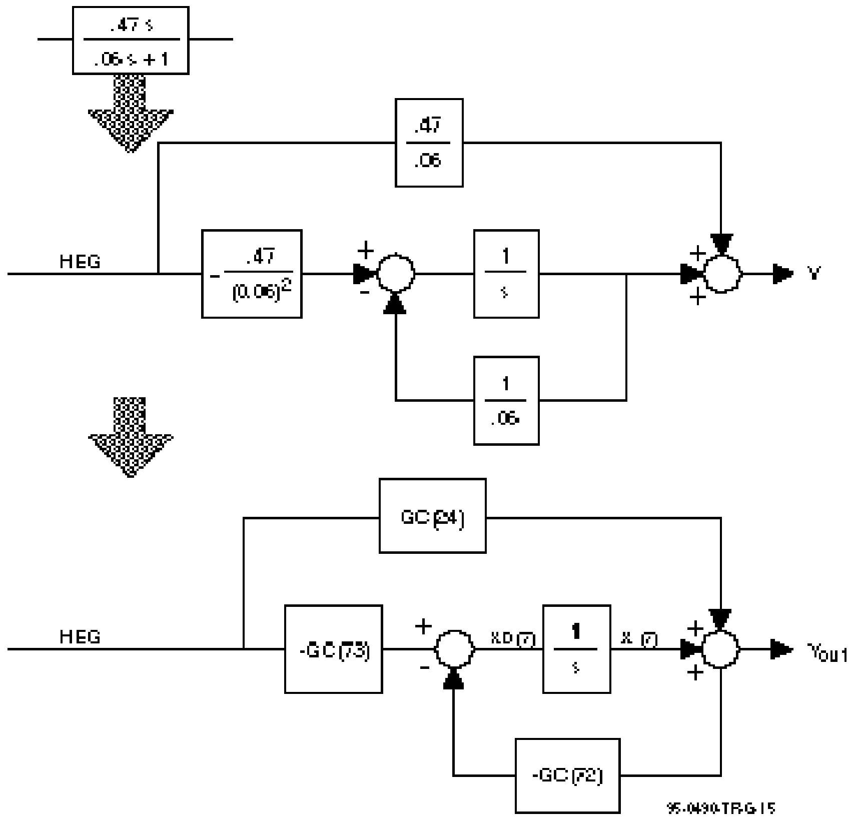
$$K_1 = b_1 - a_1 K_0 = 11.83$$

$$K_2 = b_2 - [a_2 K_0 + a_1 K_1] = 5917.16 - 1091.99 = 4824.799$$

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -5917.16 & -92.307 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} 11.83 \\ 4824.799 \end{bmatrix} u$$

SEE CLASSIFIED ADDENDUM

SEE CLASSIFIED ADDENDUM



The output of the lead net is given by:

$$Y_{out} = GC(24)\ HEG + XI(7)$$

Using the classical three point command ($R_m \theta_{TM}$) and appropriate compensation techniques and filtering, the injection mode brings the missile close to the LOS. The guidance routines in ESAMS use a standard fourth order Runge-Kutta integration algorithm. As identified in reference 4, the form is as follows.

$$Y_{n+1} = Y_n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6}, \text{ for } \varepsilon \quad \Delta x^5$$

$$k_1 = \Delta x f(x_n, y_n)$$

$$k_2 = \Delta x f\left(x_n + \frac{\Delta x}{2}, y_n + \frac{k_1}{2}\right)$$

$$k_3 = \Delta x f\left(x_n + \frac{\Delta x}{2}, y_n + \frac{k_2}{2}\right)$$

$$k_4 = \Delta x f(x_n + \Delta x, y_n + k_3)$$

In obtaining an updated value of a variable, four estimates of the increment in the variable during the program time step are obtained. The estimates are weighted as shown above, with two estimates made at the mid-point of the time step, and two at the end point.

Design Element 31-2: Low Altitude Mode

There are two primary modes: **SEE CLASSIFIED ADDENDUM**

SEE CLASSIFIED ADDENDUM

SEE CLASSIFIED ADDENDUM

FIGURE 2.31-9. Elevation Bias.

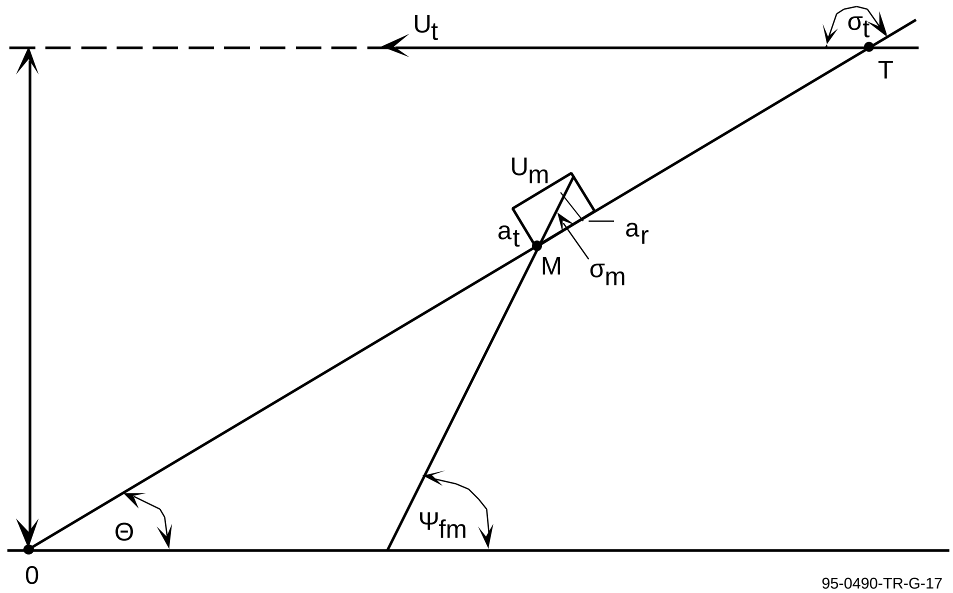
As shown in figure 3.2-1, there is a condition for the

SEE CLASSIFIED ADDENDUM

SEE CLASSIFIED ADDENDUM

Design Element 31-3: Dynamic Compensation

The requirement for dynamic compensation is due to the curvilinear motion of the three point guidance trajectory. As shown in figure 2.31.2.3-1, a body on a curvilinear trajectory experiences both radial (a_r) and transverse (a_t) accelerations. The radial component is not of much interest since it is along the flight path. However, the transverse component is of interest, since the magnitude of this component must be developed in the guidance command to keep the missile on the desired course.



RADIAL: $\frac{d^2r}{dt^2} - r\left(\frac{d\Theta}{dt}\right)^2$

TRANSVERSE: $r\frac{d^2\Theta}{dt^2} + 2\frac{dr\,d\Theta}{dt\,dt}$

Resolve into normal components to develop latex demands

$$f = R_M\,\ddot{\Theta}/COS\sigma_M + 2u_M\dot{\Theta} - \dot{u}_M SIN\sigma_M/COS\sigma_M$$

FIGURE 2.31-10. Accelerations Acting on Beam Rider Trajectory.

Reference (2) has a good discussion of this topic. As illustrated in figure 2.31.2.3-1, there are actually three components that are developed transversely. The most common one dealt with in beam rider missiles is the coriolis component ($2U_M\dot{\theta}$) with U_M being missile velocity and $\dot{\theta}$ being target flight path angular rate.

Since the missile velocity vector U_M is at an angle σ_m to the flight path, and since the missile lateral acceleration is developed normal to the missile velocity vector, the acceleration guidance command to offset the coriolis component is developed as follows:

$$a_{coriolis} = 2(U_M \cos\sigma_M)(\theta/\cos\sigma_m) = 2U_M\dot{\theta}$$

A body traveling on a curving trajectory also experiences a centrifugal acceleration $R_M\ddot{\theta}$. To offset this component, a guidance command can be used to develop the desired lateral acceleration

$$a_{centrifugal} = R_M \ddot{\theta} / \cos\sigma_M$$

Once again, there is the need for the $\cos\sigma_M$, since the desired acceleration is perpendicular to the flight path while the developed acceleration is normal to missile heading.

A final component is due to the missile acceleration not being aligned with the flight path. This feature generates an acceleration normal to the flight path of $\dot{U}_M \sin\sigma_M$. Since it is opposite in direction to the other components, the acceleration command to counter it is

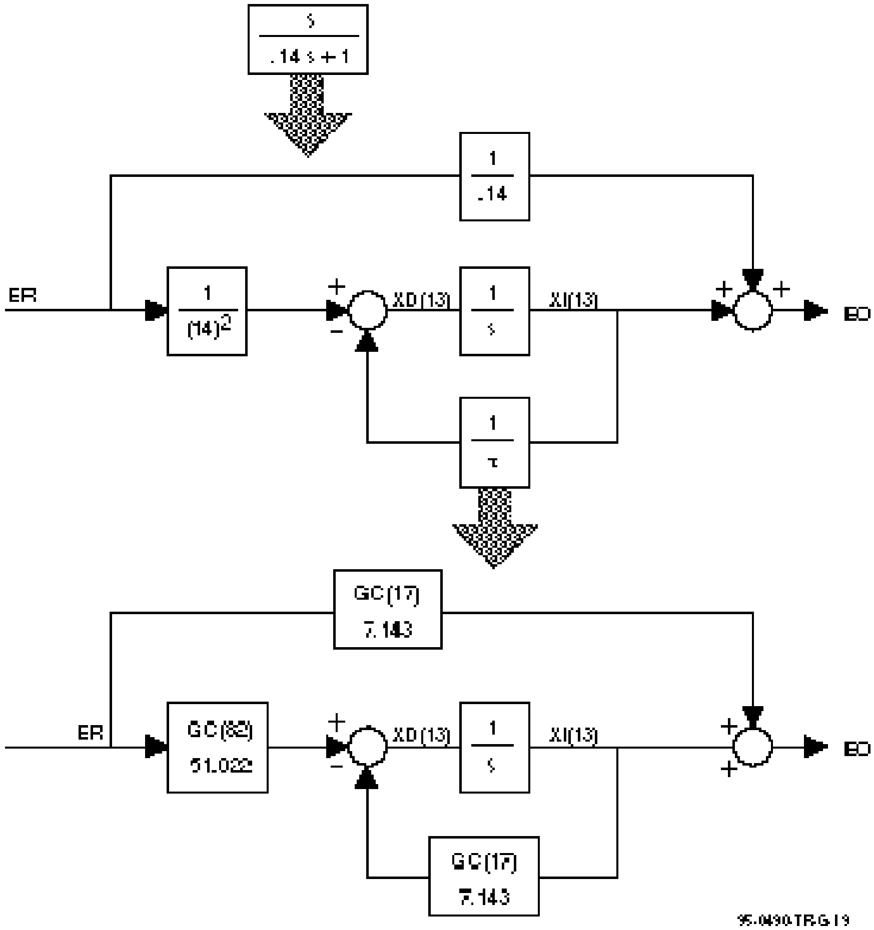
$$a_{thrust} = -\dot{U}_M \sin\sigma_M / \cos\sigma_M$$

where $\dot{U}_M$ is missile acceleration.

How accurately these components must be compensated for is dependent on several issues. If a missile has a big warhead or large inner lead zone, then compensation won't be as important. Also, tradeoffs must be made. If a parameter such as $\dot{U}_M$ is hard to acquire, then that component will probably be disregarded. Thus, tradeoffs and approximations must be used to meet system objectives.

The dynamic compensation for the elevation channel is shown in figure 2.31.2.3-2, and there are similar considerations for azimuth.

SEE CLASSIFIED ADDENDUM



$$XD(13) = -GC(17)\ XI(13) + GC(82)\ ER$$

$$EO = GC(17)\ ER + XI(13)$$

SEE CLASSIFIED ADDENDUM:

This leads to the following relations

SEECLASSIFIEDADDENDUM

EECLASSIFIEDADDENDU

SEECLASSIFIEDADDENDUM

SEECLASSIFIEDADDENDUM

ECLASSIFIEDADDENDU

SEECLASSIFIEDADDENDUM

SEECLASSIFIEDADDENDUM

SEECLASSIFIEDADDENDUM

The key relationships are

SEE CLASSIFIED ADDENDUM.

Design Element 31-4: Guide Mode

When the missile gets close to **SEE CLASSIFIED ADDENDUM**

The transfer functions in the guide circuit are represented as follows in the code.

SEECLASSIFIED decomposes in a similar manner.

The last transfer function in the guide circuit is:

$$\frac{-FKT}{2} - \frac{2V_o}{s}$$

where FKT is a missile range term. This, the first term is a direct RM θ_{TM} term. The second term integrates to be proportional to RM $\theta_{TM,}$ and the integration increases sensitivity. Thus, this furnishes a proportional plus integral feature similar to that shown in Garnell and East (page 161) and redrawn here.

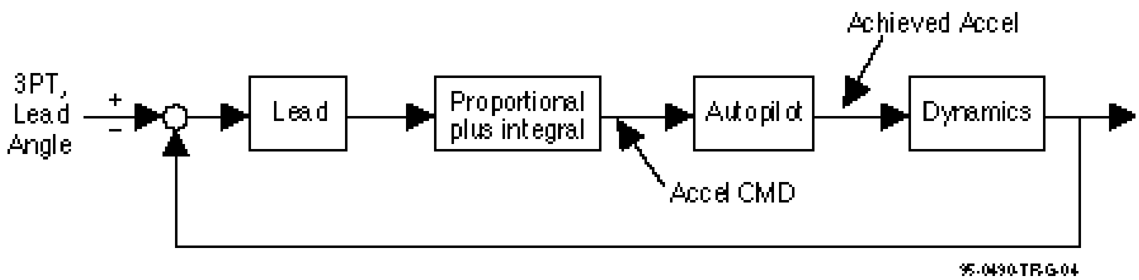


FIGURE 2.31-11. High Fidelity Command Guidance Model

The result of the lead and the proportional plus integral circuitry is to have a system with both the desired transient and steady state response. Figure 2.31.2.4-3 shows the response of the Garnell and East loop to a step input. This exhibits good missile performance.

FIGURE 2.31-12. Garnell and East Loop Response.

2.31.3 Functional Element Software Design

This section describes the software design necessary to implement the functional element requirements for command guidance, as outline in section 2.31.1 and the design approach as outlined in section 2.31.2. Section 2.31.3 is organized in four subparts: The first subpart gives the overall subroutine hierarchy and gives capsule descriptions of the relevant subroutines; the second subpart contains a functional flow chart for the functional element as a whole, and describes the major operations represented by each block in the chart; the third subpart presents detailed logical flow charts for the subroutines; and the last subpart contains a description of all input and output for the functional element as a whole and for each subroutine that implements the functional element.

Command Guidance Subroutine Hierarchy

SEE CLASSIFIED ADDENDUM

SEE CLASSIFIED ADDENDUM

TABLE 2.31-1. SEE CLASSIFIED ADDENDUM.

Module Name	Description
CONSYS	Selects system-specific guidance control routine ("GAP" routine" according to the value of the guidance type index IGTYP.
GAP8	A specific guidance control routine that sets up guidance integration control variables, calls the integrator, and converts the resulting fin deflections to voltages for use by the autopilot for this system.
RK4G	General fourth-order Runge Kutta integration routine that integrates the guidance computer state equations one time step, using the guidance integration control variables and the guidance derivative routine ("DRVG" routine) specified by the GAP routine.
RELTGT	Computes the target-missile relative geometry at the integration time-points for RK4G.
DRVG8	System-specific guidance derivative routine that implements computation of the system's guidance computer state variables and the resulting fin deflections.

Command Guidance Functional Flow

Figure 2.31.3-2 shows the top-level functional flow of the Command Guidance implementation. The actual guidance circuitry is fully contained in subroutine DRVG8. Hence, its flow is captured in figure 2.31.3-2.

SEE CLASSIFIED ADDENDUM

FIGURE 2.31-13. Command Guidance Functional Flow Diagram.

Subroutine Flow Charts

SEE CLASSIFIED ADDENDUM

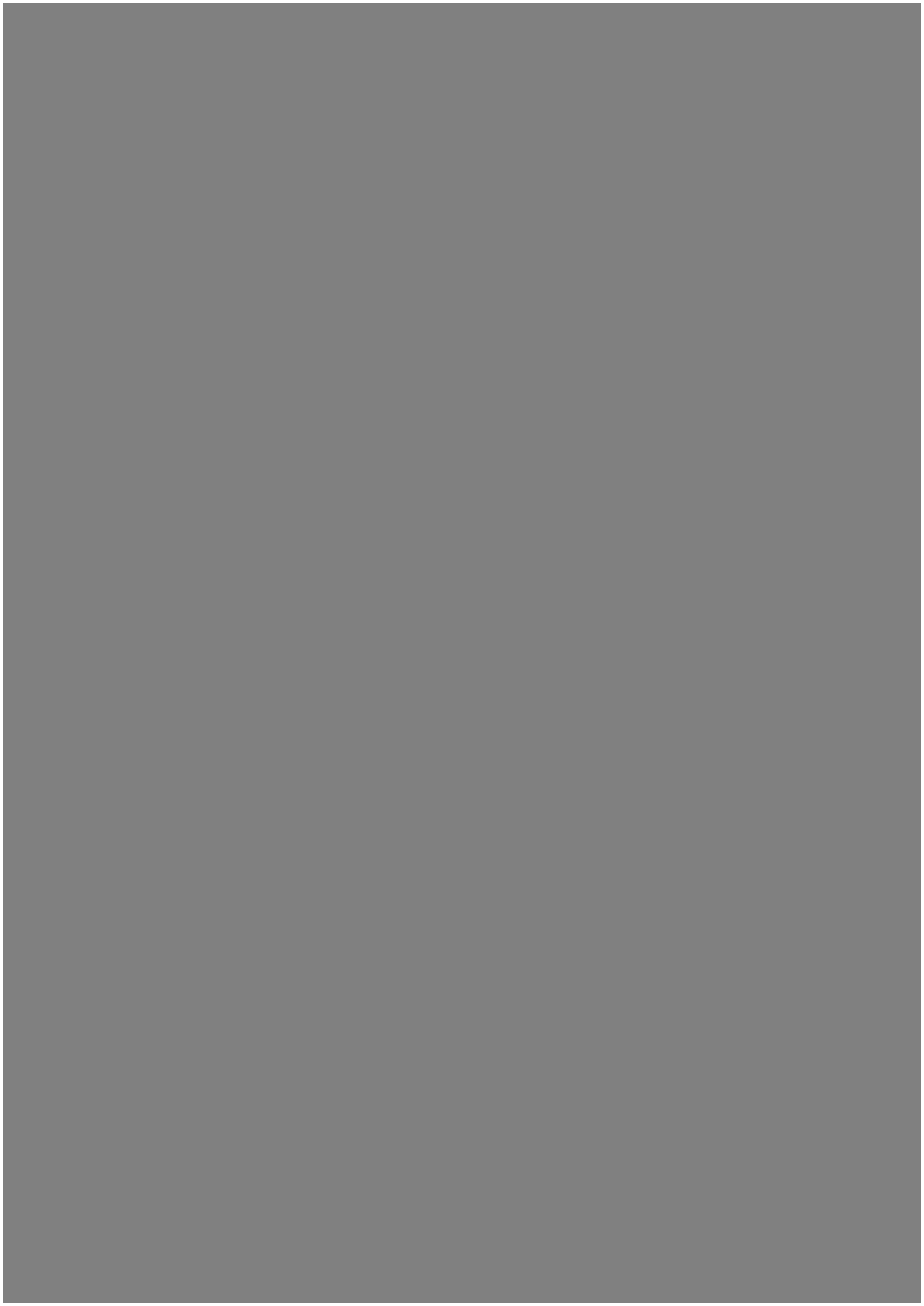


FIGURE 2.31-14. Functional Flow Diagram for Subroutine DRVG8.

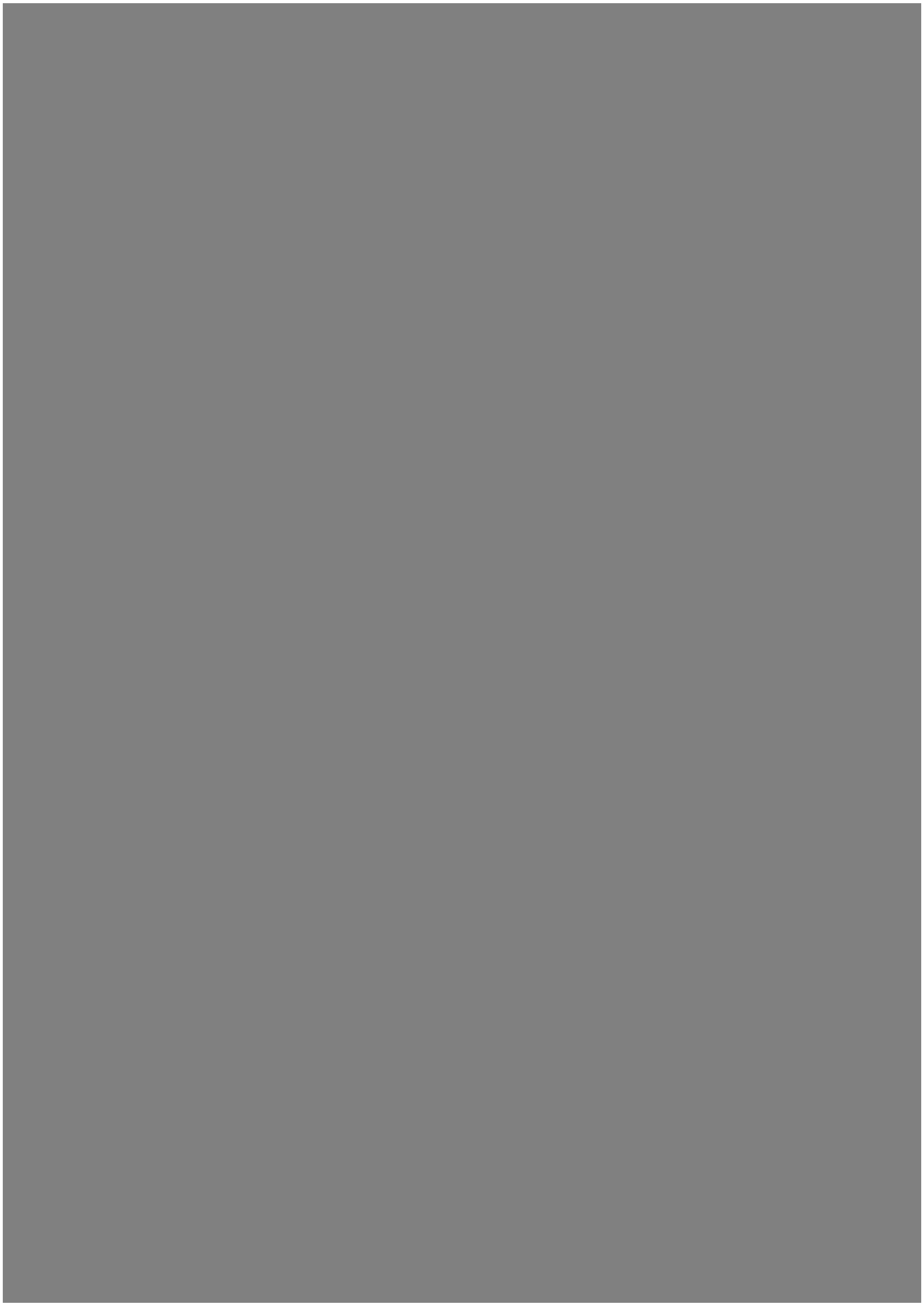


FIGURE 2.31-14. Functional Flow Diagram for Subroutine DRVG8. (Contd.)

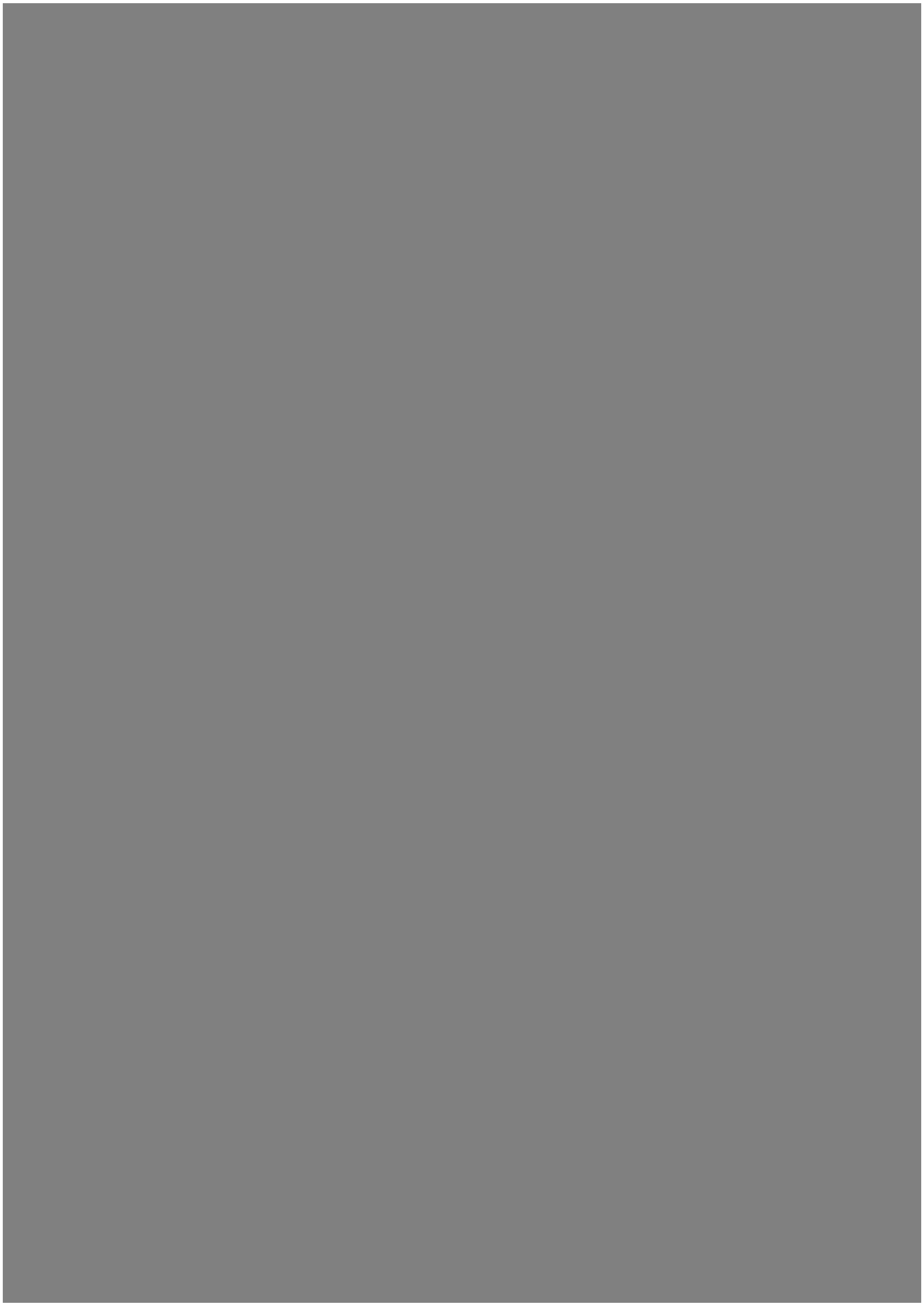


FIGURE 2.31-14. Functional Flow Diagram for Subroutine DRVG8. (Contd.)

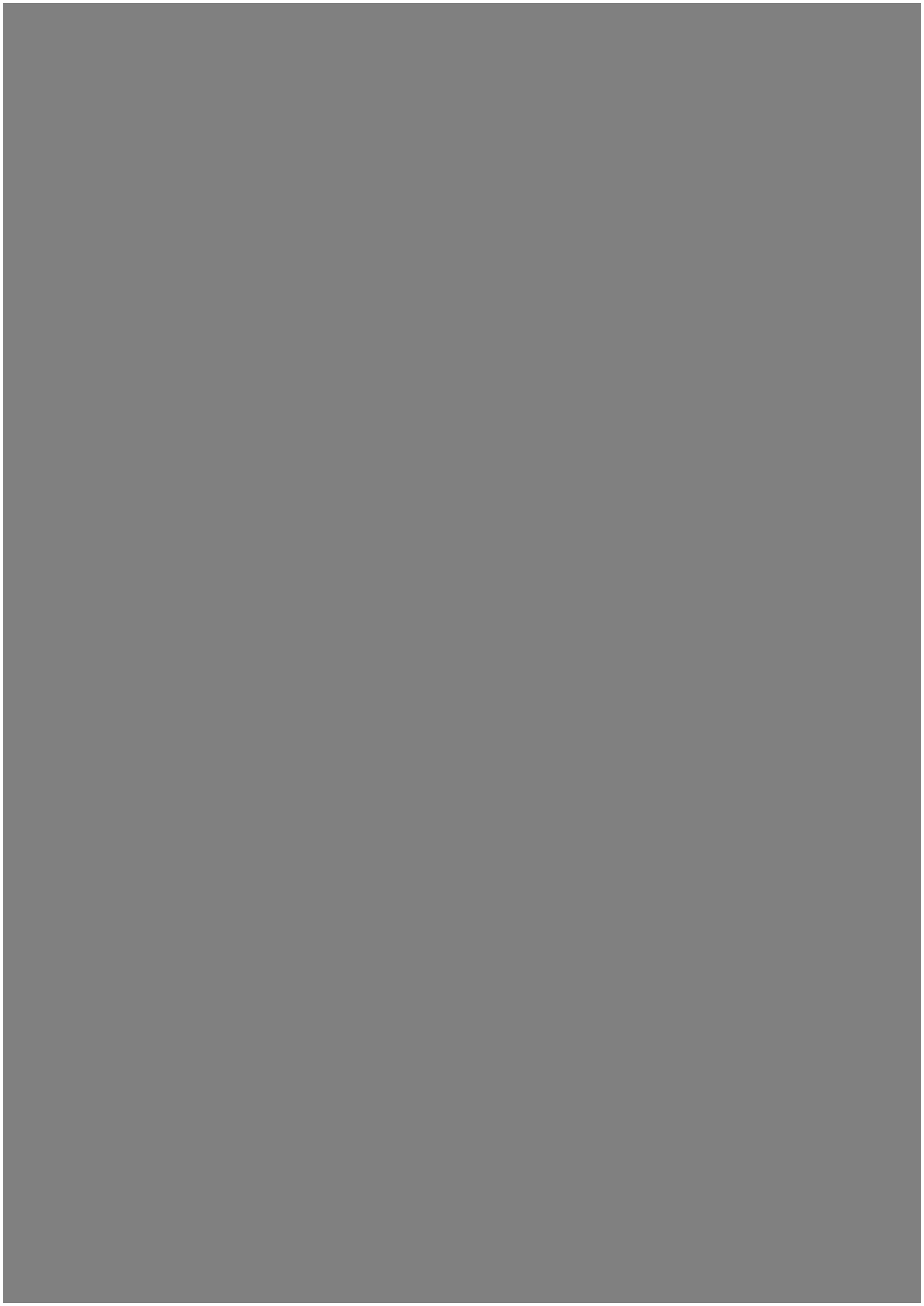


FIGURE 2.31-14. Functional Flow Diagram for Subroutine DRVG8. (Contd.)

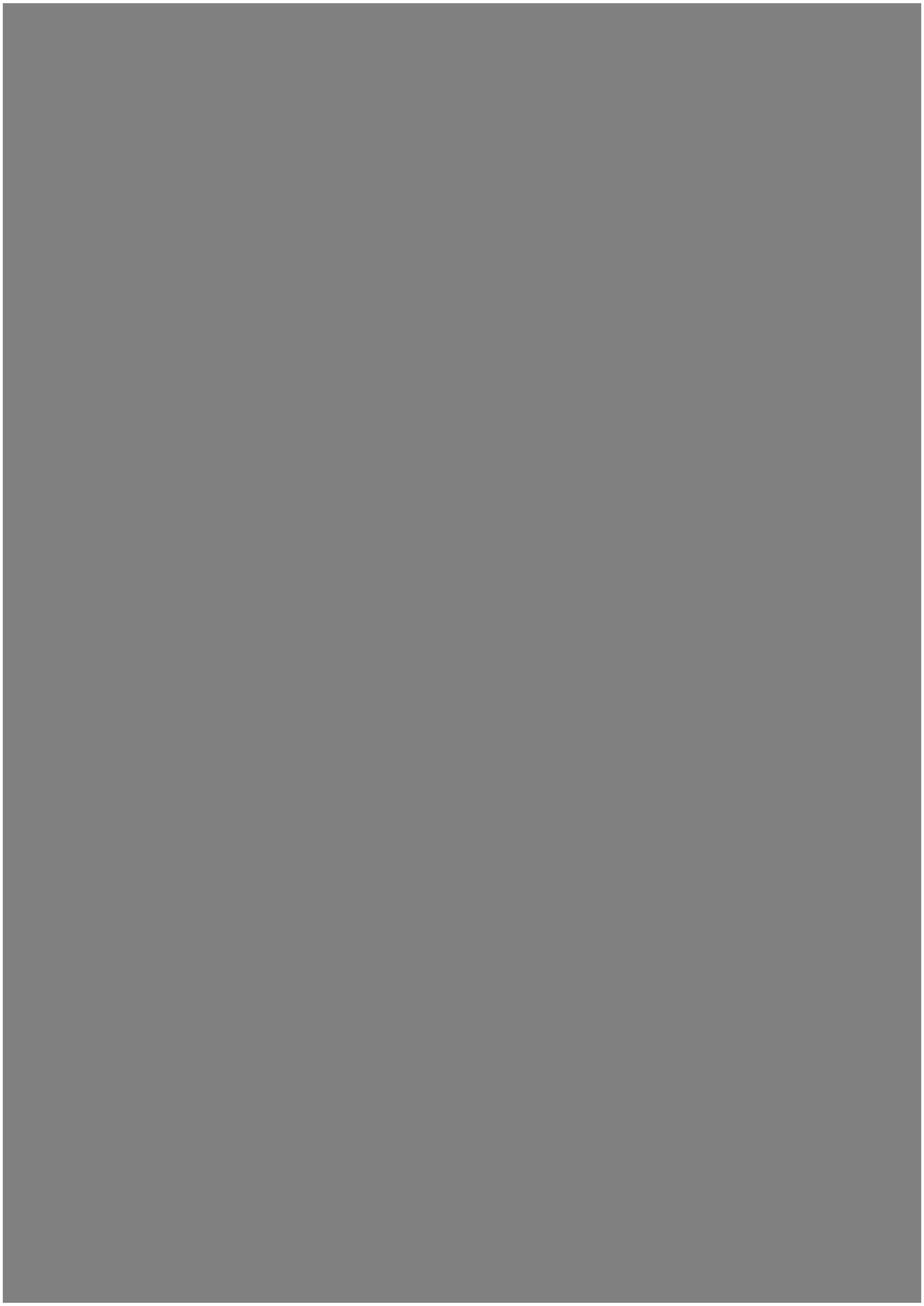


FIGURE 2.31-14. Functional Flow Diagram for Subroutine DRVG8. (Contd.)

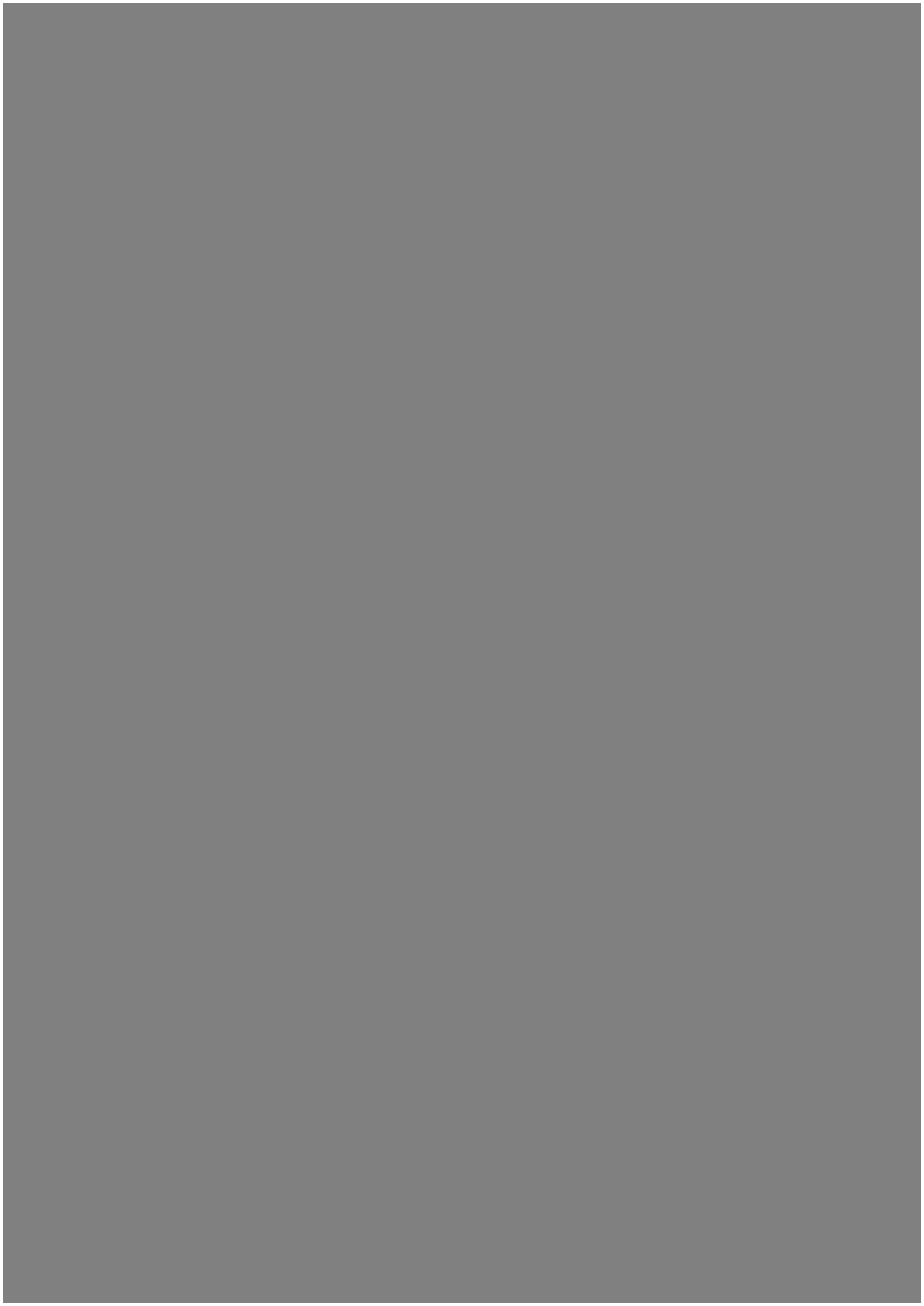


FIGURE 2.31-14. Functional Flow Diagram for Subroutine DRVG8. (Contd.)

Command Guidance Inputs and Outputs

SEE CLASSIFIED ADDENDUM

TABLE 2.31-2. Inputs of the Command Guidance Functional Element.

Name	Kind	Description
AC(-)	Common MSLD	Array of "Autopilot Coefficients". The AC(1) element is used to convert fin deflections in linear measure (meters) to voltage measure as needed by the autopilot: for most systems, this is done not in the guidance "GAP" routine, but in the autopilot "PILOT" routine.
DT	Common SIMVI	Length of time step for integration. Initialized from MSLD input DTM (missile time step) in subroutine SAMS.
GC(-)	Common MSLD	Array of guidance coefficients; meaning of individual elements depends on specific system. See prologue of subroutine DRVG8 for meanings specific to this system.
IECM	Common ROPTN	ECM mode type jamming option flag. Initialized from PROGC input RECM in subroutine PROGI.
IGTYPE	Common ROPTN	Index indicating which guidance type to use.Initialized from MSLD input FLGMSL(3) in subroutine SAMI.
IOPT	Common ROPTN	Optics option flag.Initialized from PROGC input ROPT in subroutine PROGI.
JMT1	Common RUNVI	Pointer to the EMT table in TAREA. Initialized from MSLD input PMT1 in subroutine SAMI.
JMT2	Common RUNVI	Pointer to the FTD table in TAREA. Initialized from MSLD input PMT2 in subroutine SAMI.
JMT3	Common RUNVI	Pointer to the FKT table in TAREA. Initialized from MSLD input PMT3 in subroutine SAMI.
JMT4	Common RUNVI	Pointer to the RRM table in TAREA. Initialized from MSLD input PMT4 in subroutine SAMI.
TAREA	Common MSLD	Table area of MSLD. Individual tables are specified by a table-specific pointer (offset) into TAREA. Pointers used in this functional element are JMT1, JMT2, JMT3, and JMT4.
TBALL	Common MSLD	Time in missile flight that the missile will go ballistic (sec).

SEE CLASSIFIED ADDENDUM

TABLE 2.31-3. Outputs of the Command Guidance Functional Element.

Name	Kind	Description
EK1	Common GUIDAP	Autopilot command in fin plane 1.
EK2	Common GUIDAP	Autopilot command in fin plane 2.
GDCMP	Common GUIDAP	Limited elevation guidance command. Not used as output in all command guidance systems; in particular, not used as output by systems implemented by GAP8/DRVG8.

TABLE 2.31-3. Outputs of the Command Guidance Functional Element. (Contd.)

Name	Kind	Description
GDCMY	Common GUIDAP	Limited azimuth guidance command. Not used as output in all command guidance systems; in particular, not used as output by systems implemented by GAP8/DRVG8.
IPMODE	Common SIMVI	Guidance mode type index; reset by DRVG8 if found to be zero.
KST	Common SIMVI	Counter of number of full time steps; incremented by DRVG8 when INTFLG is one.
XD	Argument	Array of derivatives to use in integration at end of current integration step; updated by the “DRVG” routine.
XI	Argument	Array of integrated variables at end of current integration step; updated by the “DRVG” routine.

SEE CLASSIFIED ADDENDUM

Input for Subroutine CONSYS		
Name	**Kind**	**Description**
FTIME	Common MISSIL	Current time from beginning of missile flight (sec).
IFATAL	PARAMETER, Include CONST	Symbolic name for value to indicate in error message that the error is a fatal kind.
IGTYPE	Common ROPTN	Index indicating which guidance type to use.
TBALL	Common MSLD	Time in missile flight that the missile will go ballistic (sec).

Output for Subroutine CONSYS		
Name	**Kind**	**Description**
APITCH	Common GUIDAP (implicit)	Current pitch acceleration command. Not used in all command guidance systems; in particular, not used by systems implemented by GAP8/DRVG8.
AYAW	Common GUIDAP (implicit)	Current yaw acceleration command. Not used in all command guidance systems; in particular, not used by systems implemented by GAP8/DRVG8.
EK1	Common GUIDAP (implicit)	Autopilot command in fin plane 1.
EK2	Common GUIDAP (implicit)	Autopilot command in fin plane 2.
ISMODE	Common SIMVI	Flag to request seeker turn on, set by guidance and used by radar logic; not used for command guidance.

Input for Subroutine GAP8		
Name	**Kind**	**Description**
AC	Common MSLD	Array of “Autopilot Coefficients”. The AC(1) element is used to convert fin deflections in linear measure (meters) to voltage measure as needed by the autopilot: for most systems, this is done not in the guidance “GAP” routine, but in the autopilot “PILOT” routine.
DT	Common SIMVI	Length of time step for integration.
EK1	Common GUIDAP	Autopilot command in fin plane 1.
EK2	Common GUIDAP	Autopilot command in fin plane 2.
FTIME	Common MISSIL	Current time from beginning of missile flight (sec).
XD	Common INTEG	Array of derivatives to use in integration at start of current time step; also updated by the calls of the “DRVG” routine.
XI	Common INTEG	Array of integrated variables at start of current time step; also updated by the calls of the “DRVG” routine.

Output for Subroutine GAP8		
Name	**Kind**	**Description**
DLT	Common SIMVI	Length of time step for integration.
DRVG8	Argument to Subroutine RK4G	EXTERNAL name of subroutine to be used as derivative support in the integration by RK4G.
EK1	Common GUIDAP	Autopilot command in fin plane 1.
EK2	Common GUIDAP	Autopilot command in fin plane 2.
N	Argument to Subroutine RK4G	Number of variables for RK4G to integrate in this case.
T	Argument to Subroutine RK4G	Time since start of missile flight at start of current integration time step (sec); is advanced by one time step by RK4G.
TWISTN	Common GUIDAP	Set in GAP8, but not actually used anywhere.

Input for Subroutine RK4G		
Name	**Kind**	**Description**
DLT	Argument	Time step for Runge-Kutta integration.
DRV	Argument	Name of “DRVG” routine to call for derivative evaluation; declared EXTERNAL.
MAXXI	PARAMETER, Include ARYBND	Dimension of intermediate integral array XIP.
N	Argument	Number of variables to integrate.
SMALL	PARAMETER, Local.	Small number used to avoid underflows.
T	Argument	Time since start of missile flight at start of current integration time step (sec).
TOL	Common SUMARY	Time of missile launch (sec).

XD	Argument	Array of derivatives to use in integration at start of current time step; also updated by the calls of the "DRVG" routine.
XI	Argument	Array of integrated variables at start of current time step; also updated by the calls of the "DRVG" routine.

Output for Subroutine RK4G		
Name	**Kind**	**Description**
T	Argument	Time since start of missile flight at end of current integration time step (sec).
XD	Argument	Array of derivatives to use in integration at end of current time step.
XI	Argument	Array of integrated variables at end of current time step.

Input for Subroutine DRVG8		
Name	**Kind**	**Description**
AZERRT	Common ENVRN	Azimuth error due to multipath and clutter.
BOREAZ	Common FREND	Target tracking radar boresight pointing angle in azimuth.
BOREEL	Common FREND	Target tracking radar boresight pointing angle in elevation.
DTR	PARAMETER, Include CONST	Degree to radian angle measure conversion factor.
ELERRT	Common ENVRN	Elevation error due to multipath and clutter.
EMT	Return from Subroutine TLU	Value of low-altitude-target elevation bias function versus time table lookup from TAREA (for pointer JMT1).
FKT	Return from Subroutine TLU	Value of angular acceleration modification function versus time table lookup from TAREA (for pointer JMT3).
FTD	Return from Subroutine TLU	Value of command generation system function versus time table lookup from TAREA (for pointer JMT2).
FTIME	Common MISSIL	Current time from beginning of missile flight (sec).
GAMMA2	Common MISSIL	Missile pitch angle (radians) of roll axis.
GC	Common MSLD	Array of guidance coefficients; meaning of individual elements depends on specific system. See prologue of subroutine DRVG8 for meanings specific to this system.
IECM	Common ROPTN	ECM mode type jamming option flag.
INTFLG	Argument	Runge Kutta integration step counter (1 to 4, corresponding to the four steps of fourth-order integration).
IOPT	Common ROPTN	Optics option flag.
IPMODE	Common SIMVI	Guidance mode type index; is reset by DRVG8 if found to be zero.
ITRKR	PARAMETER, Include ARYBND	Identifier of the tracker radar type.
JMT1	Common RUNVI	Pointer to the EMT table in TAREA.
JMT2	Common RUNVI	Pointer to the FTD table in TAREA.
JMT3	Common RUNVI	Pointer to the FKT table in TAREA.
JMT4	Common RUNVI	Pointer to the RRM table in TAREA.
KST	Common SIMVI	Counter of number of full time steps; incremented by DRVG8 when INTFLG is one.

Input for Subroutine DRVG8		
Name	**Kind**	**Description**
LMT11	Return from Subroutine TLU	Value of table lookup index from last call to TLU for the EMT table in TAREA.
LMT21	Return from Subroutine TLU	Value of table lookup index from last call to TLU for the FTD table in TAREA.
LMT31	Return from Subroutine TLU	Value of table lookup index from last call to TLU for the FKT table in TAREA.
LMT41	Return from Subroutine TLU	Value of table lookup index from last call to TLU for the RRM table in TAREA.
PI	PARAMETER, Include CONST	Mathematical constant, pi.
PIX2	PARAMETER, Include CONST	Mathematical constant, 2 times pi.
R2D	PARAMETER, Include CONST	Mathematical constant, conversion from radians to degrees.
RD2RML	PARAMETER, Include CONST	Mathematical constant, conversion from radians to Russian mil (measure of angle).
RMS	Common RELSIT	Range (slant) from site to missile.
RRM	Return from Subroutine TLU	Value of range function versus time table lookup from TAREA (for pointer JMT4) used in computing low altitude target elevation bias.
RTM	Return from Subroutine RELRV	Range between missile and target at current time.
RTMDOT	Return from Subroutine RELRV	Range rate between missile and target at current time.
RTS	Common RELSIT	Range (slant) to target from site.
T	Argument	Time since start of missile flight at start of current integration time step (sec); is advanced by one-half and one time step by RK4G.
TAREA	Common MSLD	Table area of MSLD. Individual tables are specified by a table-specific pointer (offset) into TAREA.
TIME	Common TARG	Current time (not flight time).
XD	Argument	Array of derivatives to use in integration at start of current integration step; is updated by the “DRVG” routine.
XI	Argument	Array of integrated variables at start of current integration step; is updated by the “DRVG” routine.
XMS	Common RELSIT	Missile-to-site separation, x-component
XT	Common TARG	Target location in inertial coordinate system, x-component
XTDOT	Common TARG	Target velocity in inertial coordinate system, x-component
YMS	Common RELSIT	Missile-to-site separation, y-component
YT	Common TARG	Target location in inertial coordinate system, y-component
YTDOT	Common TARG	Target velocity in inertial coordinate system, y-component
ZMS	Common RELSIT	Missile-to-site separation, z-component
ZT	Common TARG	Target location in inertial coordinate system, z-component
ZTDOT	Common TARG	Target velocity in inertial coordinate system, z-component

Output for Subroutine DRVG8		
Name	**Kind**	**Description**

EK1	Common GUIDAP	Autopilot command in fin plane 1.
EK2	Common GUIDAP	Autopilot command in fin plane 2.
GDCMP	Common GUIDAP	Limited elevation guidance command.
GDCMY	Common GUIDAP	Limited azimuth guidance command.
IPMODE	Common SIMVI	Guidance mode type index; reset by DRVG8 if found to be zero.
KST	Common SIMVI	Counter of number of full time steps; incremented by DRVG8 when INTFLG is one.
XD	Argument	Array of derivatives to use in integration at end of current integration step; updated by the “DRVG” routine.
XI	Argument	Array of integrated variables at end of current integration step; updated by the “DRVG” routine.

2.31.4 Assumptions and Limitations

SEE CLASSIFIED ADDENDUM

No limitations are known at this time.

2.31.5 Functional Process Description

2.31.5.1 Overview of Module/Subroutine Functionality

Figure 2.31.5.1-1 displays an overview of the functionality that determines the results and impact of the **SEE CLASSIFIED ADDENDUM**

FIGURE 2.31-15. SEE CLASSIFIED ADDENDUM

Discussion of the Implementation of Design Elements

Since the Command Guidance functional element implementation details are localized in the DRVG routine, which is discussed in detail in section 2.31.6.2, no additional discussion is given in this section.

Relationship of F.E. to the Whole Model

SEE CLASSIFIED ADDENDUM

2.31.6 Annotated Code

2.31.6.1 Module Description

SEE CLASSIFIED ADDENDUM

Subroutine RK4G

This routine calls an external derivative routine for four derivative evaluations. It calls RELTGT before the calls to DRV at the beginning, mid point, and end point of the current time step to update the missile-target relationship. It combines the four derivatives by the Range Kutta scheme to perform the integration.

Subroutine RELTGT

This routine will call target to update the target position, velocity, and attitude. Then it will compute a new position and velocity of the missile and new target-missile-radar site relationship.

Subroutine DRVG8

DRVG8 first updates missile position and calculates simulated radar angular measurement. Then the sensor angular outputs are differentiated and filtered to determine angular rates. An approximation to closing velocity is calculated and used to calculate guidance commands which are fed through a filter to obtain pitch and yaw commanded accelerations.

SEE CLASSIFIED ADDENDUM